Artificial Intelligence in Air Conditioner to Reduce Electricity Consumption and Maximize User Comfort

By: Lau Ying Yeung

Supervisor: Professor Chen, Chun

Final Year Project Interim Report

Submitted in Partial Fulfillment of the

Requirements for

The Degree of Bachelor of Engineering

Department of Mechanical and Automation Engineering

The Chinese University of Hong Kong

December 2019

Contents

1 Problem description

Due to an increased requirement for user comfort and convenience, smart appliances are all over the globe. At the same time, different environmental topics raised the importance of energy-efficient appliances. By combining both aspects, it becomes a trend that smart appliances should be able to improve our quality of life and cost less damage to the environment. However, there are two main reasons which prevent smart appliances from integrating into a normal household. Firstly, the price is usually higher than traditional appliances. Secondly, users do not need to replace their appliances immediately. Therefore, instead of replacing the traditional appliance, by adding a device between the user and the appliance which aims to control more precisely and tactical could be a solution to deal with the transition period of the traditional appliance to smart appliance.

The purpose for this device is to transform a traditional remote of an air conditioner (AC) to a smart remote by adding an artificial intelligence (AI) function into its design, so that it can automatically set the temperature of the AC. There are three technical problems in this device which need to be tackled. Firstly, an algorithm needs to be developed so that the remote could automatically figure out an optimal control tactic which minimize the user's discomfort and the energy usage. Secondly, it needs to distinguish the user's preferences on temperature settings. As different users may have different preferences due to their sensual feelings, the algorithm needs to find the user's control pattern. Thirdly, it requires different hardware and software components to construct the remote.

2 Result of literature review

2.1 Infrared communication

To tackle the replacement of remote, we need to understand the working principal behind the remote. The control method home appliances use is called the infrared (IR) communication. The IR frequency could lie in the range of 3kHz to 300GHz [1], but most of the infrared remote controls facilitate a modulated square wave which is between 32kHz to 40kHz [2]. The basic circuit involves three main components which are an IR Receiver, an IR LED and a development board (e.g. Arduino board, NodeMCU, etc.). To send the data to the appliance, the data is needed to encode, and it will be transmitted to the receiver by blinking the IR LED according to the pulse distance from the encoded data [3]. There are different protocols for encoding the data, NEC Infrared Transmission Protocol is one of the common protocol which uses pulse distance to encode the message bits. [4]

2.2 Supervised learning

To tackle the learning problem, we need to find out how the supervised learning could classify different things due to the given learning data sets. To train the neural network, we need to create a data sets which involves the inputs and the output of the data. After that, we need to find our hypothesis h_θ by applying sigmoid function and weight function. [5]

For linear decision boundaries,

$$hypothesis, h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots)$$

For non-linear decision boundaries,

$$hypothesis, h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 {x_1}^2 + \theta_4 {x_2}^2 \ldots)$$

$$sigmoid\ function, g(z) = \frac{1}{1+e^{-z}}$$

After finding the hypothesis, we could minimize logistic regression cost function, and apply gradient descent to find the optimal solution.

$$cost\ function, J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta\left(x^{(i)}\right) + \left(1-y^{(i)}\right) \log\left(1-h_\theta\left(x^{(i)}\right)\right)\right]$$

, which m is the total number of data sets.

Repeat {

$$\theta_j \coloneqq \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

}

, which α is the learning rate.

2.3 Reinforcement learning

To tackle the learning problem, we need to know more about reinforcement learning (RL). RL aims at a desired goal by following each decision the agent make [6]. The Markov Decision Process (MDP) is an important mathematical framework for solving RL problem [6]. There are five elements in MDP, which are the set of states, the set of actions, the transition probability, the reward probability and the discount factor. In order to motivate the agent to learn the optimal solution, a reward is given after the agent takes an action, and it will always try to maximize the total amount of rewards R_t [6].

$$\text{Total amount of rewards } R_t = r_{t+1} + r_{t+2} + \cdots + r_T$$

$$\text{Total amount of rewards with discount factor } R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \textstyle\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

Discount factor γ is used to determine whether the agent focus on future rewards or immediate rewards [6]. State value function shows the expect return when the agent is at the state s with policy π. State-action value function, which is also the Q function, shows the expect return when the agent take action a at state s.

$$state\ value\ function, V^\pi(s) = \mathbb{E}_\pi[R_t | s_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right]$$

$$state - action\ value\ function\ Q^\pi(s,a) = \mathbb{E}_\pi[R_t | s_t = s, a_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right]$$

Besides, Bellman Equation is used to find the optimal policies.

$$optimal\ \ V^*(s) = max_a \left(\sum_{s'} \mathcal{P}^a_{ss'} \left[R^a_{ss'} + \gamma Q^\pi(s', a'') \right] \right)$$

$$optimal\ \ Q^*(s,a) = max_a \left(\sum_{s'} \mathcal{P}^a_{ss'} \left[R^a_{ss'} + \gamma \sum_{a'} \pi(s', a')\, Q^\pi(s', a'') \right] \right)$$

3 Approach to the problem

3.1 Divide and conquer

In this project, as the main problem consists of different modules and topics, we could ease the problem by using the divide and conquer methods. By conquering different subproblems recursively, we combine the solutions from each sub problems to get the ultimate solution [7]. Firstly, the main problem "developing a device that can set temperature and save energy" could be divided into two subproblems which are "developing an AI control algorithm" and "developing a control device" (**Error! Reference source not found.**). Secondly, "developing an AI control algorithm" consists of "collecting inputs for the AI to learn" and "developing AI learning algorithm". While the control device consists of "replacing the air conditioner remote" and "apply AI decision to the device".

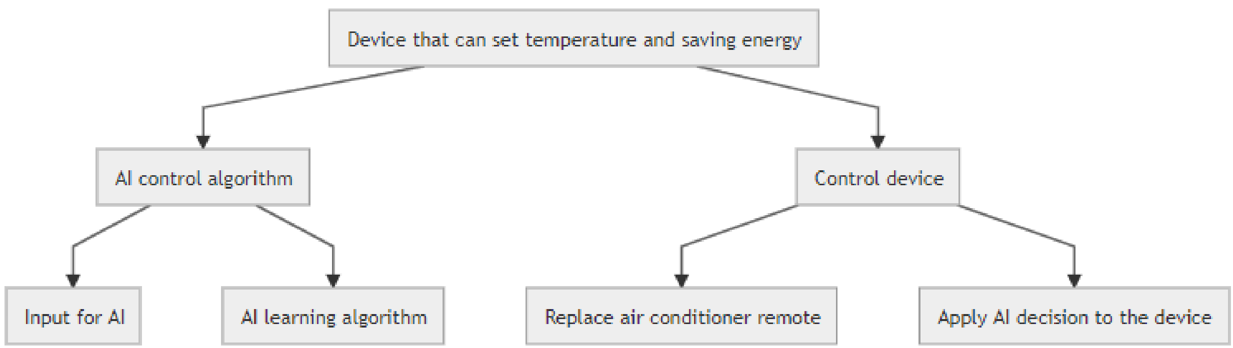

Figure 3-a Main Problem Division

From the **Error! Reference source not found.** to **Error! Reference source not found.**, the subproblems are divided into smaller subproblems which will be solved accordingly. The following sections will show different methods to solve different subproblems.

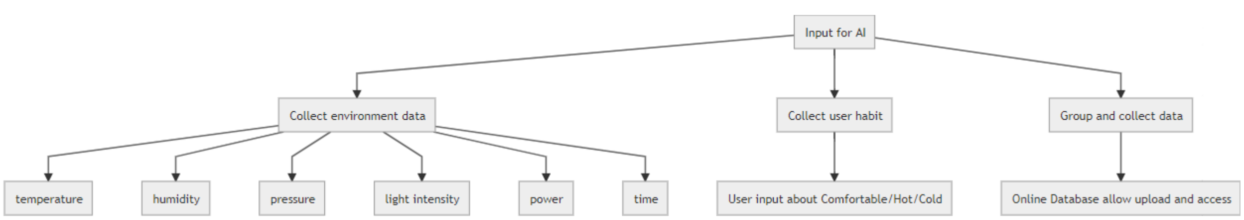

Figure 3-b Subdivision of Input for AI

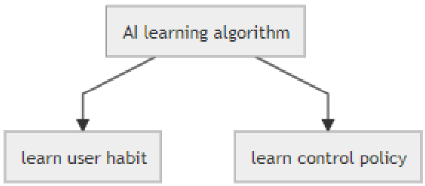

Figure 3-c Subdivision of AI learning Algorithm

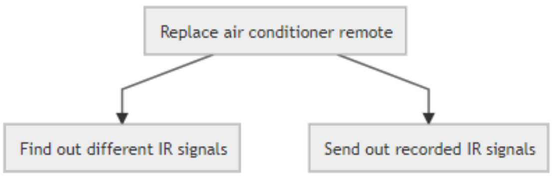

Figure 3-d Subdivision of replacing AC remote

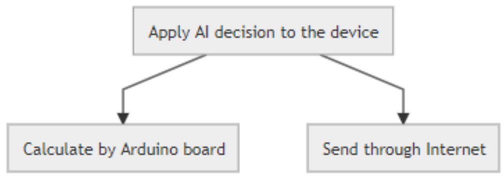

Figure 3-e Subdivision of applying AI decision to the device

3.2 AI control algorithm

3.2.1 Inputs for AI to learn

The inputs for the AI to learn consists of a set of relatable environment data and the user input. It helps the AI learn about how user like to control their AC, and the inputs could act as different states in the reinforcement learning program. The collections of data are crucial to both parts of the learning algorithm.

3.2.1.1 Collection of environment data

To collect the temperature, humidity, pressure and light intensity, we can use different sensors and an Arduino development board to read and access the values. There are three sensors which can be used to measure the four environment properties which are BMP180, BH1750 and HTU21D (Figure 3-f to Figure 3-h). To collect the power data of the AC and access the values, we need to have a power meter module that supports measuring alternating current circuit. The PZEM-004T module is used in this part (Figure 3-i). It consists of two parts which are the board and the current sensor. For the board, it is used to measure the alternating voltage. And the current sensor is used to measure the alternating current. To collect the time, we can simply use a time module for record the time of each data.

Figure 3-f BMP180 for collecting barometric pressure

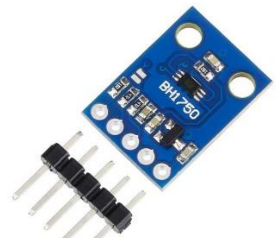

Figure 3-g BH1750 for collecting light intensity

Figure 3-h HTU21D for collecting temperature and humidity

Figure 3-i PZEM-004T and current sensor for collecting power of the AC

3.2.1.2 Collection of user input

Collecting the user input, which is about whether the user feels comfort, hot or cold, could help create data set of the user's preferences on temperature setting. To achieve this, we could use three buttons for the user to input his feedback. By connecting the three buttons to the Arduino development board, we could simply program each button for different meaning. Two types of button working principle could be selected, which are active-low and active-high (Figure 3-j) [8].

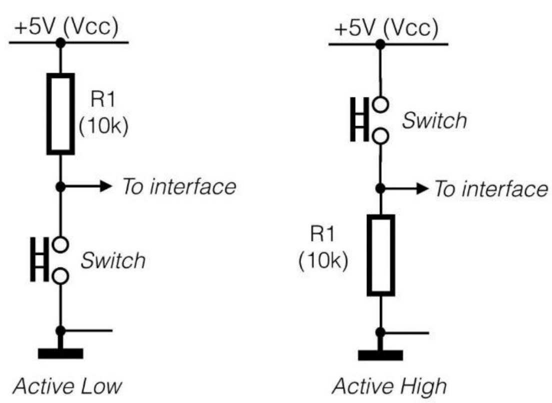

Figure 3-j Configuration of Active Low and Active High

3.2.1.3 Group and collect data

As there are a set of environment data and user input that need to be processed by the AI algorithm, it is necessary to organize and re-group the data. In order to organize the data, the environment data and user input should be recorded at a certain period, which is 1 minute in this project. As the indoor temperature takes time to change by the AC, 1 minute should be an appropriate period for recording. As the user may not have the time to input every minute, and also it is not user-friendly, it is important that we count the missing input as another feedback which is similar to comfortable, but not the same level as the user input, and we call it "acceptable". Therefore, we currently split the user feedback into four states, which are hot, cold, acceptable and comfortable.

Besides, we need to group the data in the program and send it to a platform or directly to the AI algorithm to analyze the data. In this project, we choose to use a platform to save the data we have, as it could keep things more organized, and the data could be reused. To send the data to platform, we first select a platform for storing the real-time data. As a result, this project uses the Firebase from Google to save the data (Figure 3-k).

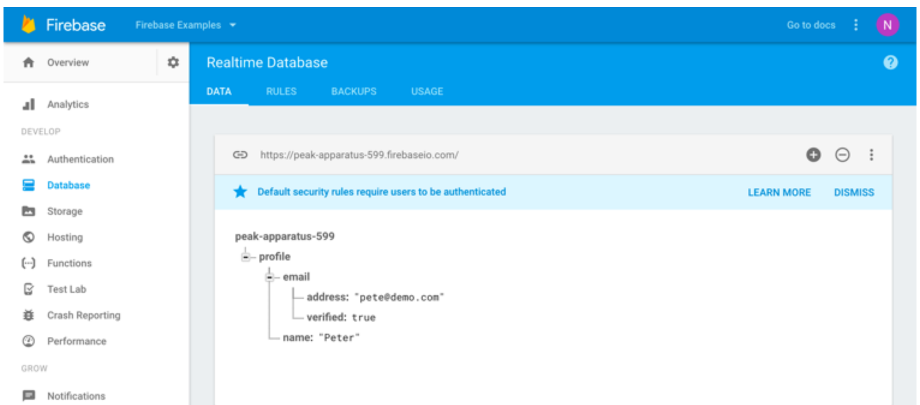

Figure 3-k Realtime database from the Firebase

3.2.2 AI learning algorithm

3.2.2.1 Learning user's habit

To figure out the user's preferences on setting the suitable temperature, we could tackle the problem by applying the supervised learning to learn the relation between the user's feedback and the environment data. For example, we know that all the environment data that will make the user feel hot, the algorithm will try to classify the unknown case that is hot or not. Therefore, learning the user's preference is a classification problem in supervised learning. For classification problem in supervised learning, it is necessary to have the labeled data for the learning process. And we need to use logistic regression instead of linear regression, as it involves using logistic function. The environment data acted as the inputs x_n to the neural network, which n is the number of the environment data, and the user input acted as the actual output y. We want our hypothesis function h_θ as close as the actual output y (Figure 3-l). Therefore, we will need to minimize the cost function and using gradient descent to find the optimal solution, in this case which is classifying the data according the user preference.

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix} \rightarrow \begin{bmatrix} a_0^{(2)} \\ a_1^{(2)} \\ a_2^{(2)} \\ \cdots \end{bmatrix} \rightarrow \begin{bmatrix} a_0^{(3)} \\ a_1^{(3)} \\ a_2^{(3)} \\ \cdots \end{bmatrix} \rightarrow \cdots \rightarrow \begin{bmatrix} h_\Theta(x)_1 \\ h_\Theta(x)_2 \\ h_\Theta(x)_3 \\ h_\Theta(x)_4 \end{bmatrix}$$

Figure 3-l The relationship between input x_n and hypothesis function h_ϑ. The hidden layer a_n is between input and output layer.

3.2.2.2 Learning control policy

To figure out the optimal control policy for the agent, we could apply the reinforcement learning to solve the problem. By giving the current state information, the agent will apply an action that maximize the return from the environment. As we need to find out the optimal policy function without having a concrete model of the environment, we are dealing with the model-free learning. And to solve this, we can use the Monte Carlo policy evaluation (Figure 3-m). By using repeat random sampling, it will keep updating the value function by the average rewards from each state. With enough repetition, the value function and policy will be updated until the change is negligible, where it is the point of convergence. Thus, we can obtain the optimal value function and optimal policy.

Initialize:
- $\pi \leftarrow$ policy to be evaluated
- $V \leftarrow$ an arbitrary state-value function
- $Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Repeat forever:
- (a) Generate an episode using π
- (b) For each state s appearing in the episode:
  - $R \leftarrow$ return following the first occurrence of s
  - Append R to $Returns(s)$
  - $V(s) \leftarrow$ average$(Returns(s))$

Figure 3-m Monte Carlo policy evaluation steps

3.3 Control device

3.3.1 Replacing air conditioner remote

By using IR receiver, we could record the control signals from the remote. We could store the signals data into array. In this project, we use the IR receiver VS1838B which can receive signals at the frequency of 38kHz. To process the signals, we need to use the IR read function from the “IRremote” library which is in the Arduino library. Also, as we need to process the signals by the Arduino board, we will need to connect the signal pin to one of the digital pins at the Arduino board.

By using IR LED, we could send the stored control signals by using the IR send function from the “IRremote” library. As the Arduino supply 5V DC, it is safer to connect a 100ohm resistor at the circuit, so that the IR LED could operate at safe voltage which is about 1.5V DC.

3.3.2 Applying AI decision to the device

There are two ways to apply the AI decision to the device, which are directly calculate by the Arduino board and send the decision through Internet. By processing the AI decision on the board, the agent could directly receive the action it takes and input the states to the Q function. However, it may not be a practical way at all time. As we need to update the Q function to the Arduino board every time the Q function is updated. Also, calculating the Q function using the Arduino board may use up the processing unit, which will make the Arduino board run slower and affect the performance. But if the decision is processed through computer, the Arduino board only need to receive the calculated decision and send the current states that the agent meets through Internet. And the requirement of the processing power will decrease, and the Arduino board could be more stable.

4 Current progress

4.1 Collecting environment data

To collect the temperature, humidity, pressure and light intensity, this device (Figure 4-a) used three integrated modules, which are BMP180, BH1750 and HTU21D. For BMP180, it collects the barometric pressure. For BH1750, it collects the light intensity from the sun. For HTU21D, it collects the indoor temperature and humidity. By using the corresponding Arduino libraries, the device can receive the signals generate by different sensors and output the correct value.

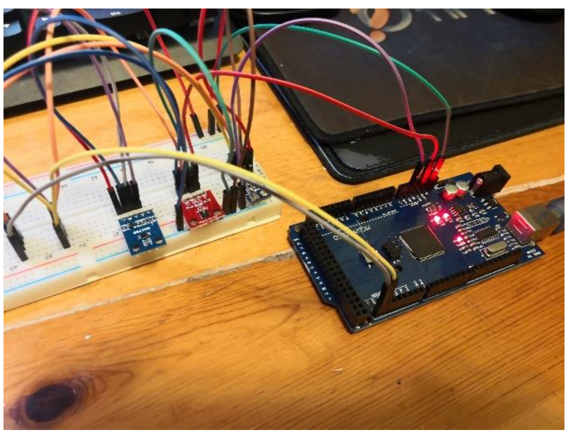

Figure 4-a connection between the three integrated modules and Arduino Mega 2560

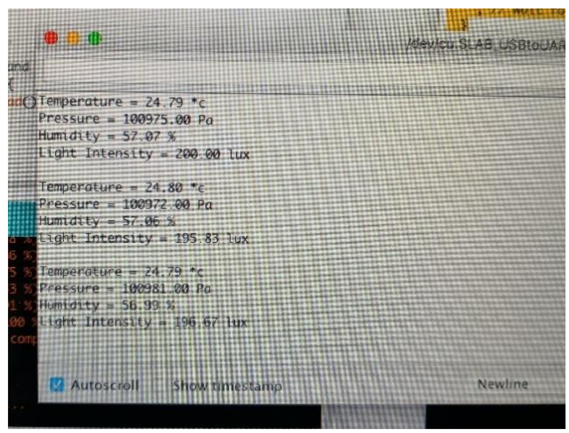

Figure 4-b the test of reading the indoor environment

4.2 IR communication

To control the air conditioner, this device used an infrared transmitter and an infrared receiver to send and receive signals. The infrared receiver, VS1838B, is specified for the frequency of 38kHz, which is a typical frequency for home appliances. The usage of the infrared receiver is to record the signals of the air conditioner remote, by recording the temperature settings signals, this device can replace the original remote, as an IR LED which is the infrared transmitter is added to this device. It is able for this device to record the infrared signals and send the recorded signals.

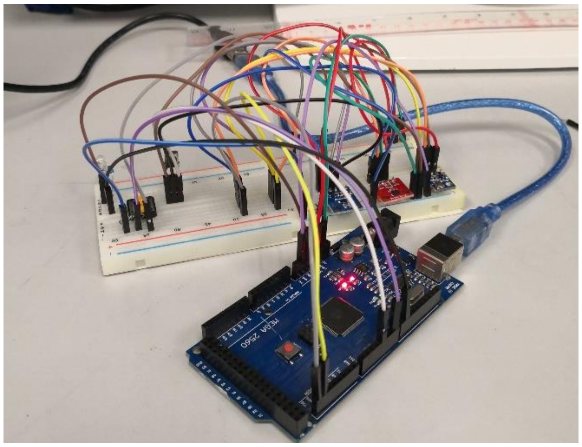

Figure 4-c connection between the infrared module and the Arduino Mega 2560

4.3 Online database connection

To record and process the data, this device will send and save the environment data to the online database (Figure 4-e). As Arduino Mega 2560 does not provide network connection, NodeMCU is used as a module for networking (Figure 4-d). The Arduino Mega 2560 will transfer the data to the NodeMCU, and the NodeMCU will send the data as a json type file to the online database directly. For the online database, this project used the Google Firebase as a online server. The online server provided a online real-time database for saving the data which is commonly used as an IoT database.

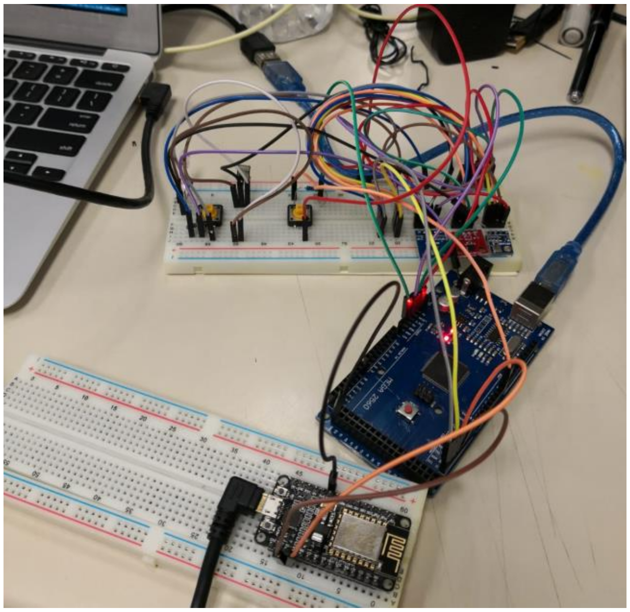

Figure 4-d connection between the Arduino Mega and NodeMCU. Smaller one is the NodeMCU.

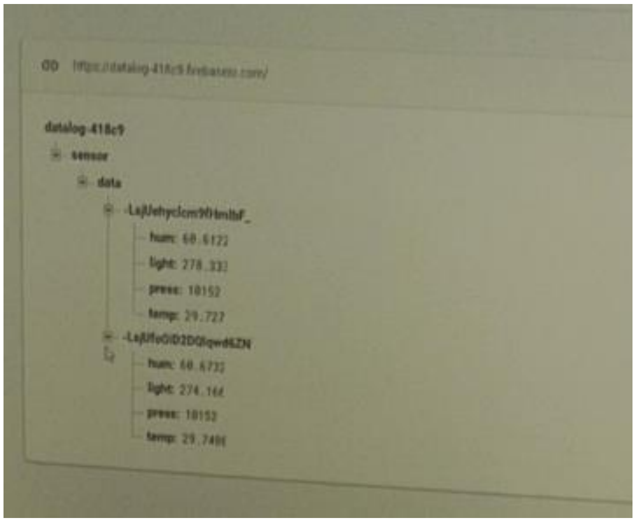

Figure 4-e Firebase receiving data

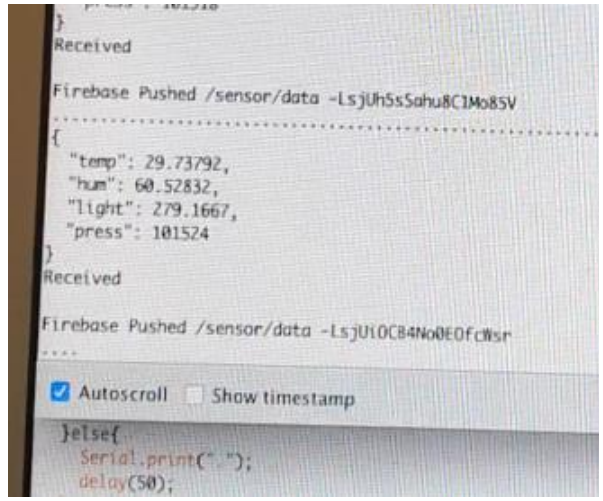

Figure 4-f Arduino board shows the data is sent to the firebase

5 List of remaining tasks

- Develop AI classification algorithm to classify the user preferences on temperature settings
- Develop AI control policy to find the optimal control policy
- Develop power meter module
- Test and improve the AI control policy

6 Work schedule for completion of remaining tasks

Tasks	Date
Develop power meter module	30/12 – 10/1
Develop AI classification algorithm	11/1 – 31/1
Develop AI control policy	1/2 – 29/2
Test and improve the AI control policy	1/3 – 3/4

7 References

[1] Ankit Jandial,Sujit Kumar,Rajat Butola,Manoj Kumar Pandey, "IR Based Home Appliances Control System," *International Journal on Recent and Innovation Trends in Computing and Communication,* vol. 5, no. 5, pp. 628-631, 2017.

[2] Santosh.M.Nejakar, "Wireless Infrared Remote Controller for Multiple," *International Journal of Electrical and Electronics Research,* vol. 2, no. 1, pp. 25-35, 2014.

[3] "IR Communication," SparkFun Electronics, 7 2 2013. [Online]. Available: https://learn.sparkfun.com/tutorials/ir-communication/all. [Accessed 29 12 2019].

[4] "NEC Infrared Transmission Protocol," Altium, [Online]. Available: https://techdocs.altium.com/display/FPGA/NEC+Infrared+Transmission+Protocol. [Accessed 13 9 2017].

[5] A. Ng, "Logistic Regression Classification," Stanford University.

[6] S. S. R. S. Y. W. Sudharsan Ravichandiran, Learning Path Python Reinforcement Learning, UK: Springer, 2019.

[7] M. Manley, "Divide and Conquer Paradigm," ACM Transactions on Computational Logic, Los Angeles.

[8] "How to connect," IPSON, [Online]. Available: http://www.ipson.nl/how-to-connect/. [Accessed 29 12 2019].